

**DETAILED ACTION**

1. This Office action is in response to the amendment filed on June 20, 2008.
2. **Claims 1-5, 7, 8, 10-25, 27, 28, and 30-41** are pending.
3. **Claims 1-5, 7, 8, 10-21, and 41** have been amended.
4. **Claims 6, 9, 26, 29, and 42-44** have been cancelled.
5. **Claims 1-5, 7, 8, 10-25, 27, 28, and 30-41** are allowed, renumbered as 1-37.

***Examiner's Amendment***

6. An Examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to Applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this Examiner's amendment was given in a telephone interview with Andrew Gabriel (Reg. No. 61,427) on October 9, 2008.

The application has been amended as follows:

**AMENDMENTS TO THE SPECIFICATION**

The amendment document filed on January 24, 2008 is considered non-compliant because it fails to meet the requirements of 37 CFR 1.121—namely, an instruction which identifies the location of an amended paragraph is incorrect. Examiner has noted this

typographical error in paragraph 5 of the Non-Final Rejection (mailed on 04/14/2008). In order for the amendment document to be compliant, please amend the incorrect instruction as follows:

On page 2 of the amendment document entitled "Amendments to the Specification," please replace the instruction "[p]lease replace the paragraph on Page 8, lines 8-12, with the following amended paragraph" with the instruction "[p]lease replace the paragraph on Page 4, lines 8-12, with the following amended paragraph."

#### AMENDMENTS TO THE CLAIMS

Please cancel Claims 42-44 and amend Claims 1-5, 7, 8, 10-21, and 41 as follows:

1. (Currently Amended) A computer-implemented method of automatically determining an allowable order of changes in a distributed system, the computer-implemented method comprising the steps of:

receiving a request for change;

wherein the request for change describes a task to be done on at least one target system, and a deadline by which a change needs to be completed;

determining existing relationship descriptions between components of the distributed system, wherein the components of the distributed system are implemented on a plurality of managed resources;

transforming acquired relationships into ordered tasks that are linked by temporal ordering constraints which describe when tasks can begin in relation to one another, wherein the

temporal ordering constraints are selected from the group consisting of: Finish-to-Start, Start-to-Start, Finish-to-Finish, and Start-to-Finish; [[and]]

creating an order of changes taking into account the temporal ordering constraints, wherein creating the order of changes includes determining whether the ordered changes are conflicting and flagging such conflicts and further includes an estimate of the time required to complete a change;

wherein the order of changes transitions the at least one target system from one workable state into another workable state[[.]];

building a Task Graph from the order of changes, wherein the Task Graph ~~which~~ represents each task to be completed within an overall job;

constructing an Annotated Task Graph by assigning estimated durations to each task within the Task Graph; and

computing a makespan for the overall job represented by the Task Graph, wherein the Annotated Task Graph is returned to an administrator.

2. (Currently Amended) The computer-implemented method of Claim 1, wherein the order of changes is sequential.

3. (Currently Amended) The computer-implemented method of Claim 1, wherein the order of changes is concurrent.

4. (Currently Amended) The computer-implemented method of Claim 1, further comprising refining an incoming request for change by breaking the incoming request down into sub-requests.

5. (Currently Amended) The computer-implemented method of Claim 4, further comprising computing an allowable order of changes by interacting with the distributed system.

7. (Currently Amended) The computer-implemented method of Claim 1, wherein the ordered changes are partially ordered.

8. (Currently Amended) The computer-implemented method of claim 1, wherein the ordered changes are totally ordered.

10. (Currently Amended) The computer-implemented method of Claim 4, wherein a total change time is minimized by exploiting parallelism between change tasks.

11. (Currently Amended) The computer-implemented method of Claim 1, wherein the creation of the order of changes further takes into account a requested change management operation.

12. (Currently Amended) The computer-implemented method of Claim 1, wherein a requester identifies one or more target systems within the distributed system by name.

13. (Currently Amended) The computer-implemented method of Claim 12, wherein the names of the target systems are unique physical identifiers.

14. (Currently Amended) The computer-implemented method of Claim 12, wherein the names of the target systems are logical names which refer to one or more physical systems.

15. (Currently Amended) The computer-implemented method of Claim 1, wherein a requester does not identify one or more target systems within the distributed system by name.

16. (Currently Amended) The computer-implemented method of Claim 1, further comprising the steps of accessing and evaluating policy roles representing best practices.

17. (Currently Amended) The computer-implemented method of Claim 16, wherein the best practices include updating all affected software artifacts when a software artifact is updated.

18. (Currently Amended) The computer-implemented method of Claim 16, wherein the best practices include having a given set of software components installed on different systems.

19. (Currently Amended) The computer-implemented method of Claim 1, wherein one or more of the order of changes are persistently stored after being created.

20. (Currently Amended) The computer-implemented method of Claim 1, wherein a component is one of a service, an application, middleware, hardware, an operating system, a storage system, a network device, and a system associated with a computing environment.

21. (Currently Amended) A system for automatically determining an allowable order of changes in a distributed system, the system comprising:

a processor; and

a memory storing code accessible by the processor to:

receiving a request for change;

wherein the request for change describes a task to be done on at least one target system, and a deadline by which a change needs to be completed;

determining existing relationship descriptions between components of the distributed system, wherein the components of the distributed system are implemented on a plurality of managed resources;

transforming acquired relationships into ordered tasks that are linked by temporal ordering constraints which describe when tasks can begin in relation to one another, wherein the temporal ordering constraints are selected from the group consisting of: Finish-to-Start, Start-to-Start, Finish-to-Finish, and Start-to-Finish; [[and]]

creating an order of changes taking into account the temporal ordering constraints, wherein creating the order of changes includes determining whether the ordered changes are conflicting and flagging such conflicts and further includes an estimate of the time required to complete a change;

wherein the order of changes transitions the at least one target system from one workable state into another workable state[.]];

building a Task Graph from the order of changes, wherein the Task Graph which represents each task to be completed within an overall job;

constructing an Annotated Task Graph by assigning estimated durations to each task within the Task Graph; and

computing a makespan for the overall job represented by the Task Graph, wherein the Annotated Task Graph is returned to an administrator.

41. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for automatically determining an allowable order of changes in a distributed system, said method comprising the steps of:

receiving a request for change;

wherein the request for change describes a task to be done on at least one target system, and a deadline by which a change needs to be completed;

determining existing relationship descriptions between components of the distributed system, wherein the components of the distributed system are implemented on a plurality of managed resources;

transforming acquired relationships into ordered tasks that are linked by temporal ordering constraints which describe when tasks can begin in relation to one another, wherein the

temporal ordering constraints are selected from the group consisting of: Finish-to-Start, Start-to-Start, Finish-to-Finish, and Start-to-Finish; [[and]]

creating an order of changes taking into account the temporal ordering constraints, wherein creating the order of changes includes determining whether the ordered changes are conflicting and flagging such conflicts and further includes an estimate of the time required to complete a change;

wherein the order of changes transitions the at least one target system from one workable state into another workable state[. . .];

building a Task Graph from the order of changes, wherein the Task Graph which represents each task to be completed within an overall job;

constructing an Annotated Task Graph by assigning estimated durations to each task within the Task Graph; and

computing a makespan for the overall job represented by the Task Graph, wherein the Annotated Task Graph is returned to an administrator.

-- END OF AMENDMENT --

***Reasons for Allowance***

7. The following is an Examiner's statement of reasons for allowance:

The cited prior art taken alone or in combination fail to teach, in combination with the other claimed limitations, "building a Task Graph from the order of changes, wherein the Task Graph represents each task to be completed within an overall job; constructing an Annotated



Task Graph by assigning estimated durations to each task within the Task Graph; and computing a makespan for the overall job represented by the Task Graph, wherein the Annotated Task Graph is returned to an administrator” as recited in independent Claims 1, 21, and 41.

The closest cited prior art, the combination of US 5,721,824 (hereinafter “Taylor”) and US 6,675,382 (hereinafter “Foster”), teaches installing a software package having at least one dependent software package also to be installed on a server or standalone file space, multiple client file space or both in the file system of the server and one or more clients. However, the combination of Taylor and Foster fails to teach “building a Task Graph from the order of changes, wherein the Task Graph represents each task to be completed within an overall job; constructing an Annotated Task Graph by assigning estimated durations to each task within the Task Graph; and computing a makespan for the overall job represented by the Task Graph, wherein the Annotated Task Graph is returned to an administrator” as recited in independent Claims 1, 21, and 41.

Any comments considered necessary by Applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

### ***Conclusion***

8. The prior art made of record and not relied upon is considered pertinent to Applicant’s disclosure.

9. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Q. C./

Examiner, Art Unit 2191

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191